SYSTEM AND METHOD FOR DETERMINING AFFIXES OF WORDS

Background Of The Invention

Field of the Invention

[0001] This invention, generally, relates to the field of computer lexical acquisition and text analysis, and more specifically, the invention relates to finding affixes (prefixes, infixes, and suffixes) in words. Even more specifically, the preferred embodiment of the present invention provides a tool for automatically collecting affixes of a language from one or more documents, and for saving those affixes in a database for future use.

Background Art

[0002] Knowledge of affixes is important for analyzing existing words and also for producing new words. This knowledge helps to improve search results and to identify newly created words in text.

[0003] Affixes make it possible to group words into a canonical form. If we have well-complied prefixes and suffixes, and if we are given the word "play," then we know "played," "playing," "plays," "replay," "replayed," and so on are variants of "play." Affixes also add meanings to existing words. For example "or" is a suffix and when it is attached to a verb, it means "a person who performs the action described by the verb." Thus, we can interpret "actor" as "act" and "or," and understand that "actor" means a person who acts. "Antiasthmatic" is composed of "anti" and "asthmatic" and means something works against "asthma."

[0004] Many never-seen words are found in text due to the advances of new technologies and the creativity of human beings. A common way of generating new words is creating morphological variations of existing words, such as *auto-inject-or* and *co-sponsor-ship*. These newly created words that are unknown to the lexicon cause many problems for Natural Language Processing (NLP) systems.

[0005] However, if we can segment a new word into the stem and the affix(es), we can obtain linguistic information about the word such as the possible parts-of-speech and the meaning. Most present systems for morphological analysis and out-of-vocabulary handling require a precompiled list of affixes and morphological rules specifying how each affix can apply. It is very difficult and time-intensive to acquire a complete list of affixes of a language by hand.

[0006] Recently, efforts have been developing for automatically identifying morphemes. For instance, such efforts are disclosed in "Discovering Morphemic Suffixes A case Study In MDL Induction," by Brent, et al, In Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, Ft. Lauderdale, Fl. (1995) (Brent, et al.); "Unsupervised Learning of Naïve Morphology with Genetic Algorithms," by Dimitar Kazakov, Pages 105 to 112, Workshop Notes of the ECML/Mlnet Workshop on Empirical Learning of Natural Language Processing Tasks, Prague, Czech Republic (April 1997) (Kazakov); "Knowledge-free Induction of Inflectional Morphologies," by Schone, et al. Proceedings of the North American chapter of the Association for Computational Linguistics (NAACL)(2001) (Schone, et al.); "Unsupervised Learning of Morphology Using a Novel Directed Search Algorthm: Taking the First Step," by Snover, et al, Association for computational Linguistics (ACL-2002): Workshop on Morphological and Phonological Learning (Snover, et al.); "Unsupervised Learning of the Morphology of a Natural Language," by John Goldsmith, Computational Linguistics, Volume 27, Number 2 (2001) (Goldsmith); and "Using Distributional Information to Discover Morphemes: An Automated Distribution-Driven Prefix Learner," by Marco Baroni, presented at the Morphology Meeting, Vienna, Austria, February 2002 (Baroni)..

[0007] More specifically, Brent, et al. describes a system aiming at finding the right set of suffixes based on Minimum Description Length (MDL). This system, though, requires a part-of-speech tagged corpus. Kazakov also describes a MDL-based suffix finding algorithm from a raw text. This algorithm uses a simple genetic algorithm with MDL as the fitness function. Goldsmith describes another MDL-based system that attempts to provide both a list of morphemes and an analysis of each word in a corpus. He considers every possible split of each word in a corpus, and uses MDL as well as triage procedures to

eliminate inappropriate parses. Schone and Jurafsky employe many sophisticated post-hoc adjustments to obtain the right conflation sets for words. Snover, et al. describes a system for the unsupervised learning of morphological suffixes and stems from a list of the most common words in a language. It tries to detect the final stem and suffix break of each word.

[0008] The first stage for discovering prefixes or suffixes would be finding a good division of a word into prefix, stem and suffix. Previous approaches parse words into two pieces: prefix and stem, or stem and suffix, depending on their goals. The most common way to parse a word in these systems is referred to as "split-all"; that is, to consider all possible splits. In this case, there are w-1 possible splits for a word of w characters.

[0009] This method is simple but computationally expensive. For instance, Kazakov reports that the computation complexity of the algorithm limits the size of the input lists to hundreds of words, and the algorithm took eight and a half hours to find suffixes from 120 words on a Pentium 90 Mhz platform.

[0010] Previous approaches limit the length of an affix to reduce the size of the search space. Those systems, otherwise, become impractical due to too many possible parses of words and complicated computations of statistical information. For instance, Goldsmith limits the length of a suffix to five letters arguing that no grammatical morphemes require more than five letters in familiar languages. However, longer prefixes and suffixes are easily found in many documents. Some examples from biomedical documents are such prefixes as *pharmaco*, *plethysmo* and *ventriculo* and such suffixes as *ectomy*, *ibility* and *ogenesis*.

[0011] Baroni aims at finding a set of prefixes from a corpus based on distributional cues, which include distribution, frequency and length of words and their substrings in the input data.

[0012] The prior art systems have a number of disadvantages and limitations. For instance, the prior art fails to discover both prefixes and suffixes at the same time, and the prior art does not automatically analyze documents to find infixes. The prior art splits a word only into two parts – "a stem" + "a suffix," or "a prefix" + "a stem." Thus, if a stem has a

prefix, the system cannot discover the prefix. Likewise, if a stem has a suffix, the system cannot find the suffix.

[0013] Also, the prior art cannot discover nested affixes (more than one prefix or suffix, such as the prefix "radio-immuno" and the suffix "less-ness"). The limitation on the length of an affix in the prior art keeps the prior art from finding many affixes that appear in technical documents, for example documents that exist in the fields of medicine, engineering, etc. In addition, the prior art fails to find affixes containing non-alphabetic characters such as digits and hyphens. These kinds of affixes often appear in technical documents such as the biomedical domain and chemistry.

Summary Of The Invention

[0014] An aspect of this invention is an improved system and method for determining affixes in one or more documents.

[0015] Another aspect of this invention is an improved system and method for determining nested affixes in one or more documents.

[0016] A further aspect of this invention is an improved system and method for determining affixes in one or more documents where the affixes are not available to the system in a dictionary.

[0017] Another aspect of this invention is an improved system and method for determining affixes in one or more documents in order to obtain stem words.

[0018] An aspect of this invention is an improved system and method for determining affixes in one or more documents to create dictionaries of affixes.

[0019] A further aspect of this invention is an improved system and method for determining affixes in one or more documents to provide input to other natural language processing systems.

[0020] The present invention provides a computer system and a method for analyzing text in one or more electronic documents. The computer system comprises one or more system interfaces; and an affix process that determines one or more affixes of one or more words in one or more of the documents and provides the affixes to the system interface.

[0021] The preferred embodiment of this invention, as described in detail below, may be used to build a domain specific morphology lexicon for NLP applications so that they can recognize out-of-vocabulary words. Furthermore, the processes of discovering prefixes and suffixes are not independent. Many words, especially in technical documents, have complex morphological structures, and thus the knowledge about prefixes helps the discovery of suffixes and vice versa. For example, for *rhino-sinus-itis*, if we knew that rhino is a prefix, then we can consider *itis* is a suffix even though *rhinosinus* does not appear in the lexicon nor in the documents.

[0022] In addition, in the preferred embodiment, the present invention does not limit the length of an affix, and is, consequently, able to find not only long affixes but also nested affixes (e.g., more than one prefix or suffix such as *radioimmuno* and *lessness*, respectively).

Brief Description Of The Drawings

[0023] The foregoing and other aspects and advantages of the invention will be better understood from the following, non-limiting, detailed description of preferred embodiments of the invention, given with reference to the drawings that include the following:

[0024] Figure 1 is a block diagram illustrating one preferred embodiment of the present invention.

[0025] Figure 2 is a flow chart of an affix finding process.

[0026] Figures 3 and 4 are diagrams showing tree representations of words in one or more document that are used to explain the affix finding process.

[0027] Figure 5 is an example output of a preferred system of this invention.

Detailed Description Of The Preferred Embodiments

[0028] Domain-specific documents contain many domain-specific morphemes, and the knowledge on the morphemes is important to process those documents. The preferred embodiment of this invention provides an unsupervised, knowledge-free procedure for automatically discovering prefixes and suffixes from text.

[0029] The preferred procedure represents words in the documents as a Patricia (Practical Algorithm to Retrieval Information Coded in Alphanumeric) tree in order to easily identify the morphological structures of words, and thus acquire better potential candidates. Also, this procedure integrates prefix and suffix discovery in such a way that it can use knowledge about prefixes to find suffixes and vice versa.

[0030] The system of the invention is very useful for building morphology lexicons for NLP systems, especially for domain-specific documents. The performance of the procedure is evaluated with a domain-specific document collection and a general document collection. In addition, the preferred procedure of this invention proves to be simpler, faster and more accurate than previous approaches.

[0031] One goal of the preferred embodiment of the invention is to automatically generate lists of possible affixes from documents. Affixes for some human languages such as English and German have been relatively well studied and collected, typically by linguists. However, this is not true for many less-studied languages and technical languages, for example, in the biomedical domain.

[0032] For instance, 150 of 219 prefixes found in a biomedical document collection are domain specific prefixes. The preferred embodiment of this invention provides an unsupervised, knowledge-free procedure to automatically discover prefixes and suffixes of a language from text.

[0033] An affix is any element in the morphological structure of a word other than a stem. Affixes are traditionally divided into prefixes, which come before the form to which

they are joined; suffixes, which come after; and infixes, which are inserted within a stem. The specific embodiment of the invention described herein focuses on finding prefixes and suffixes, but the present invention could be used for identifying infixes as well.

[0034] For the sake of simplicity, a word may be considered as being composed of zero or one prefix, a stem, and zero or one suffix. When a word has nested prefixes or suffixes, it is not necessary to try to separate these affixes into two affixes. For example, *radioimmuno* may be considered as a prefix and *lessness* may be considered as a suffix in *radioimmuno-assay* and *sleep-lessness* respectively.

[0035] More formally, prefixes and suffixes of a language L may be defined as follows: A string **p** is a prefix of a word **w** if **w = pw'** for another word **w'** of the language. A string **s** is a suffix of a word **w** if **w = w"s** for another word **w"** of the language.

[0036] Figure 1 generally illustrates the operation of a preferred embodiment of the present invention. Generally, the affix finding system or procedure, referenced at 100, searches through documents, which may be documents 105 available on the Internet, for new affixes. When these affixes are found, they are stored in storage medium 195. These new affixes may then be used by computer applications 200 in a search through documents and/or speeches 300. All of this may be done under the control or supervision of an operator, via a computer 400. Also, that computer 400 may have access to an electronic dictionary 500 to assist in the search for the affixes.

[0037] Figure 2 is a flow chart illustrating steps of procedure 100. On-line documents are represented at 105, and at step 110, the next word to be studied is obtained. At step 120, the word is added into a prefix Patricia tree; while at steps 130 and 140, the word is reversed and then the reversed word is added into a suffix Patricia tree. At step 145, a decision is made whether more words are to be added to either of the Patricia trees. If so, the procedure returns to step 110 and continues on from there. Steps 110, 120, 130, 140 and 145 may be repeated until no more words are to be added to the Patricia trees.

[0038] Then, from step 145, the procedure goes to steps 150, 160 and 170. At step 150, all potential affixes are generated; at step 160, the number of good stems for affixes are counted; and at step 170, the affixed stems are processed. After step 170, the procedure goes to step 180, where affixes are disambiguated; and, after this, new affixes are generated at step 190. These new affixes are then added to storage medium 195.

[0039] The preferred procedure of this invention first represents all the words in the collection as Patricia trees, which visually show the morphological structures of the words and enables the potential candidates of prefixes and suffixes to be easily identified.

[0040] A Patricia tree is a compact representation of a trie. However, when constructing a trie over a large number of and extremely long strings, many of the internal nodes in the trie have only one descendent, and thus waste memory. A trie is turned into a Patricia tree by compressing all unary paths. Leaves and nodes with more than one child remain and represent the same string they did in trie.

[0041] Furthermore, the preferred procedure integrates prefix and suffix discovery so that it can use knowledge about prefixes to find suffixes and vice versa. It improves the candidate prefixes and suffixes through iterative refinement.

[0042] The preferred embodiment of the invention generates the initial sets of potential prefixes and suffixes from Patricia trees, and gradually refines the set of prefixes or suffixes with the knowledge on suffixes or prefixes which was discovered in the previous steps.

[0043] Figures 3 and 4 show examples of Patricia trees. More specifically, Figure 3 is an example of a prefix Patricia tree for the words {anti-anxiety, anti-cancer, antioxidative, cardioacceleratory, cardioactive, cardiography, neurocognitive, neurologist}. In this example, all the strings for the internal nodes are potential prefixes. Figure 4 is an example of a tree for reverse words, and in particular, for the reverse of the words {anti-anxiety, anti-cancer, antioxidative, cardioacceleratory, cardioactive, cardiography, neurocognitive, neurologist}. In this tree, all the strings for the internal nodes are potential suffixes.

[0044] Figure 5 shows an example output that may be obtained with the present invention. This output was obtained searching through biomedical documents for prefixes and suffixes in a collection of "medline" abstracts. The upper half of Figure 5 lists the prefixes that were found, and the lower half of the Figure lists the suffixes that were found.

[0045] The preferred embodiment of the invention, as described above, has a number of important advantages. For example, by using a Patricia tree, this preferred embodiment can recognize potential prefixes or suffixes without parsing words into pieces. A Patricia tree visually shows common substrings of words, and these common substrings are actually potential affixes. Also, the preferred embodiment does not limit the length of an affix, and is able to find not only long affixes but also nested affixes. In addition, the present invention may be effectively used to find affixes containing non-alphabetic characters such as digits and hyphens.

[0046] While it is apparent that the invention herein disclosed is well calculated to fulfill the objects stated above, it will be appreciated that numerous modifications and embodiments may be devised by those skilled in the art, and it is intended that the appended claims cover all such modifications and embodiments as fall within the true spirit and scope of the present invention.